*The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.*

**TLP: WHITE**
**Disclosure is not limited. Subject to standard copyright rules, TLP: WHITE information may be distributed without restriction.**
**http://www.us-cert.gov/tlp/**

**DATE(S) ISSUED:**
08/10/2020

**SUBJECT:**
Multiple Vulnerabilities in PHP Could Allow for Arbitrary Code Execution

**OVERVIEW:**
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow for arbitrary code execution. PHP is a programming language originally designed for use in web-based applications with HTML content. PHP supports a wide variety of platforms and is used by numerous web-based software applications. Successful exploitation of the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.

**THREAT INTELLIGENCE:**
There are currently no reports of these vulnerabilities being exploited in the wild.

**SYSTEMS AFFECTED:**
- PHP 7.2.31 prior to version 7.2.32
- PHP 7.3 prior to version 7.3.20
- PHP 7.4 prior to version 7.4.8

**RISK:**
**Government:**
- Large and medium government entities: **High**
- Small government entities: **High**

**Businesses:**
- Large and medium business entities: **High**
- Small business entities: **High**

**Home users: Low**

**TECHNICAL SUMMARY:**
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow an attacker to execute arbitrary code. Details of these vulnerabilities are as below:

Version 7.4.9

- Fixed bug #79030 (Upgrade apache2handler's php_apache_sapi_get_request_time to return usec).
- Fixed bug #63208 (BSTR to PHP string conversion not binary safe).
- Fixed bug #63527 (DCOM does not work with Username, Password parameter).
- Fixed bug #79740 (serialize() and unserialize() methods can not be called statically).
- Fixed bug #79783 (Segfault in php_str_replace_common).
- Fixed bug #79778 (Assertion failure if dumping closure with unresolved static variable).
- Fixed bug #79779 (Assertion failure when assigning property of string offset by reference).
- Fixed bug #79792 (HT iterators not removed if empty array is destroyed).
- Fixed bug #78598 (Changing array during undef index RW error segfaults).
- Fixed bug #79784 (Use after free if changing array during undef var during array write fetch).
- Fixed bug #79793 (Use after free if string used in undefined index warning is changed).
- Fixed bug #79862 (Public non-static property in child should take priority over private static).
- Fixed bug #79877 (getimagesize function silently truncates after a null byte) (cmb)
- Fixed bug #79756 (finfo_file crash (FILEINFO_MIME)).
- Fixed bug #55857 (ftp_size on large files).
- Fixed bug #79787 (mb_strimwidth does not trim string).
- Fixed bug #79797 (Use of freed hash key in the phar_parse_zipfile function). (CVE-2020-7068)
- Fixed bug #79487 (::getStaticProperties() ignores property modifications).
- Fixed bug #69804 (::getStaticPropertyValue() throws on protected props).
- Fixed bug #79820 (Use after free when type duplicated into ReflectionProperty gets resolved).
- Fixed bug #70362 (Can't copy() large 'data://' with open_basedir).
- Fixed bug #78008 (dns_check_record() always return true on Alpine).
- Fixed bug #79839 (array_walk() does not respect property types).

Version 7.3.21

- Fixed bug #79030 (Upgrade apache2handler's php_apache_sapi_get_request_time to return usec).
- Fixed bug #79877 (getimagesize function silently truncates after a null byte).
- Fixed bug #79778 (Assertion failure if dumping closure with unresolved static variable).
- Fixed bug #79792 (HT iterators not removed if empty array is destroyed).
- Fixed bug #63208 (BSTR to PHP string conversion not binary safe).
- Fixed bug #63527 (DCOM does not work with Username, Password parameter).
- Fixed bug #79741 (curl_setopt CURLOPT_POSTFIELDS asserts on object with declared properties).
- Fixed bug #79756 (finfo_file crash (FILEINFO_MIME)).
- Fixed bug #55857 (ftp_size on large files).
- Fixed bug #79787 (mb_strimwidth does not trim string).
- Fixed bug #79797 (Use of freed hash key in the phar_parse_zipfile function). (CVE-2020-7068)
- Fixed bug #70362 (Can't copy() large 'data://' with open_basedir).
- Fixed bug #79817 (str_replace() does not handle INDIRECT elements).
- Fixed bug #78008 (dns_check_record() always return true on Alpine).

Version 7.2.33

- Fixed bug #79877 (getimagesize function silently truncates after a null byte) (cmb)
- Fixed bug #79797 (Use of freed hash key in the phar_parse_zipfile function). (CVE-2020-7068)

Successful exploitation of the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.

**RECOMMENDATIONS:**

The following actions should be taken:

- Upgrade to the latest version of PHP immediately, after appropriate testing.
- Verify no unauthorized system modifications have occurred on system before applying patch.
- Apply the principle of Least Privilege to all systems and services.
- Remind users not to visit websites or follow links provided by unknown or untrusted sources.

**REFERENCES:**

**PHP:**

https://www.php.net/ChangeLog-7.php#7.4.9
https://www.php.net/ChangeLog-7.php#7.3.21
https://www.php.net/ChangeLog-7.php#7.2.33